REMARKS

Below, the applicant's comments are preceded by related remarks of the examiner set forth in small bold type.

3. Claim 1, 2, 4-8, 10, 11, 13-17, 29-32 and 34 are rejected under 35 U.S.C. 103(a) as being unpatentable over Johnson et al. (US 2003/0084253) in view of Ebner et al. (2003/0105929).

Regarding Claims 1-2, 34, Johnson discloses a computer system comprising: a cache (Figure 1) including cache lines to store data (lines 0-3 of data section 106), at least a portion of the data to be written to main memory (when a cache line is evicted, it is copied to or written to main memory, paragraph 006), and an eviction mechanism (state machine 116, Figure 1 and paragraph 0010) to evict data stored in one of the cache lines (cache line not accessed or changed... may be preemptively evicted based on validity state information (age bit in second logical state) associated with the data stored in the cache line, the eviction mechanism to send evicted data to the main memory (when a cache line is evicted, it is copied to main memory, paragraph 006). Johnson does not disclose each of a subset of the cache lines having multiple portions and validity information that indicates the status of the data in respective portion in the cache line. Ebner discloses Valid Portion bits 211 (see Figure 2) indicating which part of the data line is valid (page 2, paragraph 25). The Valid portion bits 211 reflect the validity of the portions (multiple portions)of the cache line (page 3, paragraph 29). It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the eviction process of Johnson to evict cache lines based on the valid portion bits of Ebner, since giving the system to ability to identify invalid portions of a flexibility to the manner in which the cache is accessed (Ebner, Abstract).

Johnson does not disclose or suggest "an eviction mechanism to evict data stored in one of the cache lines upon detecting validity bits indicating that respective portions of the cache line have been written with new write data," as recited in amended claim 1.

Johnson discloses "age-bits [that] are used to indicate whether lines in the cache may be stale." (paragraph 0017, emphasis added). Johnson's cache system does not evict data in a cache line "upon detecting validity bits indicating that respective portions of the cache line have been written with new write data," but rather, waits for a predetermined amount of time and evicts the data from the cache line when an age-bit indicates that the data are stale. (paragraphs 0020 and 0021)

Ebner discloses a "Valid portion 211" that includes "bits indicating which part[s] of the data is valid." (paragraph 0025). Ebner does not explain when data would be considered "valid,"

but even if Ebner's valid bits could be used to indicate whether "respective portions of the cache line have been written with new write data," there is no suggestion or motivation to replace Johnson's age-bits with the Ebner's valid bits as the trigger for data eviction. Ebner itself does not disclose any criteria for eviction of data in a cache line. The combination of Johnson and Ebner merely suggests using valid bits to indicate which parts of the cache line is valid, and then evicting the data in the cache line when the data are stale as indicated by the age-bit.

Claims 2, 4-9, 27, 29, and 34 are patentable for at least the same reasons as claim 1.

Regarding Claim 10, Johnson discloses a computer system comprising: cache lines to store bytes of data that correspond to consecutive addresses in a main memory (Figure 1, lines 0-3, data section 106) at least a portion of the data to be written to the main memory (data is written to main memory when evicted, paragraph 006) each cache line corresponding to a group of validity bits (age bits, paragraph 0010) each of the validity bits tracking a portion of the cache line and being set to a predefined value when the tracked portion of the cache line is fully written in one write transaction (set to a first logical state); and an eviction component to evict the bytes of data stored in one of the cache lines when the group of validity bits corresponding to the cache line are all set to the predefined value (second logical state), the eviction component to send the evicted data to the main memory (evicted data is copied to main memory, paragraph 006). Johnson does not disclose each of a subset of the cache lines having multiple portions and validity information that indicates the status of the data in respective portion in the cache line. Ebner discloses Valid Portion bits 211 (see Figure 2) indicating which part of the data line is valid (page 2, paragraph 25). The Valid portion bits 211 reflect the validity of the portions of the cache line (page 3, paragraph 29). It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the eviction process of Johnson to evict cache lines based on the valid portion bits of Ebner, since giving the system to ability to identify invalid portions of a cache line allows the system to be more precise in its determination of what valid resources are available for use, thus adding flexibility to the manner in which the cache is accessed (Ebner, Abstract).

Although Johnson discloses initializing age-bits to a first logical state, Johnson does not disclose or suggest "a validity bit that is set to a predefined value when the corresponding portion of the cache line is fully written with new data in one write transaction, the validity bit not being set to the predefined value if the corresponding portion of the cache line is not fully written with new data or if the corresponding portion of the cache line is fully written with new data in two or more write transactions," as recited in amended claim 10.

What is lacking in Johnson is also not disclosed or suggested in Ebner, which discloses "if the requester intends to write to a portion of the cache line, the Valid Portion bits 211 would be required to be changed to reflect the validity of the portion to be written to." (paragraph 0025)

In addition, neither Johnson nor Ebner discloses or suggests "an eviction component to evict the bytes of data stored in one of the cache lines when validity bits corresponding to the multiple portions of a cache line are all set to the predefined value," for reasons discussed above for claim 1.

Claims 11, 12, and 30 are patentable for at least the same reasons as claim 10.

Regarding Claims 13-14, Johnson discloses the method of a computer system comprising: receiving write transactions associated with data to be written to a main memory wherein the write transactions into the cache are received when a cache miss occurs and the missed data is written to the cache memory, assuming the this data will be accessed again soon (paragraph 006); storing the data into portions of a single cache line of a cache, and evicting the write data from the cache line when the cache line is full of data (if a cache is full, a new line must replace an existing line... if the replaced line is dirty, the line must be evicted) according to stored validity information (dirty bit). The evicted data is copied to the main memory. Johnson does not disclose each of a subset of the cache lines having multiple portions and validity information that indicates the status of the data in respective portion in the cache line. Ebner discloses Valid Portion bits 211 (see Figure 2) indicating which part of the data line is valid (page 2, paragraph25). The Valid portion bits 211 reflect the validity of the portions of the cache line (page 3, paragraph 29). It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify the eviction process of Johnson to evict cache lines based on the valid portion bits of Ebner, since giving the system to ability to identify invalid portions of a cache line allows the system to be more precise in its determination of what valid resources are available for use, thus adding flexibility to the manner in which the cache is accessed (Ebner, Abstract).

Johnson and Ebner, independently or in combination, do not disclose or suggest "evicting the write data from the cache line upon detecting validity bits indicating that respective portions of the cache line have been fully written with new write data," for at least reasons similar to claim 1.

Claims 14-17 are patentable for at least the same reasons as claim 13.

4. Claims 18-19, 21-23, 28, and 33 rejected under 35 U.S.C. 103(a) as being unpatentable over Johnson et al. (US 2003/0084253) in view of Percival (US 2004/0186958).

Regarding Claim 18, Johnson discloses a computer apparatus comprising: a computer chipset comprising a cache memory to store write data (Figure 1) and a mechanism (state machine) to evict the write data from the cache memory when a set of predefined conditions are met (age bits in second logical state, paragraph 010). Johnson

evicts the write data from the cache when the cache line is full of data according to stored validity information (dirty bit). The evicted data is copied to the main memory. Johnson does not disclose a cache memory to store data from an input/output device. Percival discloses a cache that caches data from at least one I/O device (Abstract and Claim 37). Since the system of Johnson is that of a common computer system, the system must inherently be connected to an input/output device given that a system such as this system is usually connected to a keyboard or mouse (input/output device), therefore, it would have been obvious to one of ordinary skill in the art to allow the cache to store I/O data as done in Percival's invention since it is known that a user has the ability to create files through the use of a keyboard and mouse (I/O device)and such files are the generated I/O data that is stored in the cache. Additionally, the write data that needs to be written to the cache must be of smaller or equal in size to the evicted data since the evicted data is evicted in order to make room to store the first data (paragraph 006).

The examiner acknowledges that Johnson does not disclose a cache memory to store data from an input/output device. The examiner points to Percival as disclosing what is missing in Johnson.

The cache in Percival does not "store write data sent from an input device," as recited in amended claim 18. What Percival discloses is a disk cache 10 for caching data from disks. (paragraph 0014) In Percival, an "io intercept global" routine is used to check I/O devices of a computer system to determine whether an I/O device is "one of the disk device types supported by the cache software of the invention." (paragraph 0023) Percival does not disclose or suggest caching data from I/O devices other than disk devices, and a disk device is not "an input device," as recited in claim 18.

Claims 19-21 and 33 are patentable for at least the same reasons as claim 18.

Regarding Claim 22, Johnson discloses the method of a computer system comprising: writing the data into a cache memory (Figure 1); evicting the data from the cache memory (when the age-bits are on second logical state, paragraph 010); and writing the data into a main memory (evicted data is copied into main memory, paragraph 006). Johnson does not teach initiating write transactions by an input/output device to write data. Percival discloses a cache that caches data from at least one I/O device (Abstract and Claim 37). Since the system of Johnson is that of a common computer system, the system must inherently be connected to an input/output device given that a system such as this system is usually connected to a keyboard or mouse (input/output device), therefore, it would have been obvious to one of ordinary skill in the art to allow the cache to store I/O data as done in Percival's invention since it is known that a user has the ability to create files through the use of a keyboard and mouse (I/O device) and such files are the generated I/O data that is stored in the cache. Additionally, the write data that needs to be written to the cache must be of smaller or equal in size to the evicted data since the evicted data is evicted in order to make room to store the first data (paragraph 006).

Johnson and Percival, independently or in combination, does not disclose or suggest "initiating write transactions by an input device to write data," as recited in amended claim 22, for at least reasons similar to claim 18.

Claims 23-26 and 28 are patentable for at least the same reasons as claim 22.

Please apply the required $790.00 Request for Continued Examination Fee and $250.00 excess claims fees and any other charges or credits to deposit account 06-1050, referencing attorney docket 10559-639001.

Respectfully submitted,

Date: 6/22/2005

Rex Huang

Rex I .Huang* for
David L. Feigenbaum, Reg. No. 30,378

Fish & Richardson P.C.
225 Franklin Street
Boston, MA 02110-2804
Telephone: (617) 542-5070
Facsimile: (617) 542-8906

** See attached document certifying that Rex Huang has limited recognition to practice before the U.S. Patent and Trademark Office under 37 CFR § 10.9(b).*

21073221.doc